Teste Cientista de Dados

Instruções básicas: Utilize um ambiente de desenvolvimento Jupyter com kernel Python 3+. Para solucionar os desafios, utilize de quaisquer bibliotecas que julgar necessário, a menos que o enunciado especifique o contrário. Fique a vontade para utilizar um ambiente de desenvolvimento local ou remoto (exemplo: Google Colab). A entrega deve ser o próprio arquivo de notebook, no seguinte padrão: nome-sobrenome.ipynb

- Sabemos que o processo de desenvolvimento pode ser demorado, e a busca de melhores parâmetros para modelos é algo que exige processamento, na prova iremos avaliar muito mais a metodologia empregada, do que o resultado do modelo em si.

1

Utilize o dataset existente em https://www.kaggle.com/fedesoriano/stroke-prediction-dataset como base para realizar a prova.

2

Realize uma análise exploratória, utilizando de gráficos e análises estatísticas, para indicar o comportamento dos dados, inclusive dados faltantes. Assuma o completo desconhecimento da base de dados e procure quaisquer pontos relevantes para uma futura etapa de modelagem.

- Não se limite nessa análises, nos mostre o que você considera importante em uma análise exploratória.

3

Realise as etapas de pré-processamento e separação dos dados que julgar adequadas para o treinamento de um modelo de classificação binária sobre a variável alvo, utilizando todas as demais colunas do DataFrame como variáveis do modelo.

4

Apresente um gráfico 2D de dispersão dos dados de teste em relação à variável alvo. Neste procedimento as colunas de variáveis devem ser transformadas apenas em duas, formando os eixos X

e Y do gráfico. Para amostras com valor de "TARGET" 0 utilizar a cor azul; já para amostras de "TARGET" 1 utilizar vermelho.

5

A partir dos dados da etapa 3, treine um modelo de classificação binária baseado em árvore. Utilize uma técnica de otimização de hiper-parametros a sua escolha.

6

A partir dos dados da etapa 3, treine um modelo de classificação binária baseado em redes neurais. Utilize uma técnica de otimização de hiper-parametros ou busca automática de rede a sua escolha.

7

A partir dos dados da etapa 3, treine um modelo utilizando qualquer técnica a sua escolha, desde que seja diferente das utilizadas nas questões 5 e 6.

8

Compare os resultados dos tres modelos acima e justifique a métrica escolhida para avaliação.

9

Escolha um dos tres modelos acima. Supondo que um falso negativo tenha um custo muito maior do que falso positivo, e sabendo que a predição da variável alvo é realizada entre 0 e 1, mesmo em modelos de classificação binária, selecione um limiar para definir se uma pessoa terá ou não um derrame e justique a sua escolha.

10

Realize uma análise de "Equal Error Rate" e uma análise "SHAP" para o modelo a cima e descreva suas conclusões.

11

Crie uma classe que carregue o modelo treinado do disco e tenha uma função de predição. A função de predição deve receber como parâmetro uma única amostra de dados e retorne o resultado de predição (0 ou 1) utilizando o limiar definido na questão 9.

12

Disserte sobre como seria o modelo ideal para que o modelo gerado seja colocado em produção, de forma que possa ser utilizado para realizar predições em tempo real.

13

Escreva uma função que receba uma lista (array) de numeros inteiros e retorne um booleano (True ou False) indicando se a lista é monotônica. Para essa não é permitido o uso de nenhuma biblioteca, somente a linguagem Python com seus tipos nativos.
def isMonotonic(A: List[int]) -> bool:
<seu código aqui>